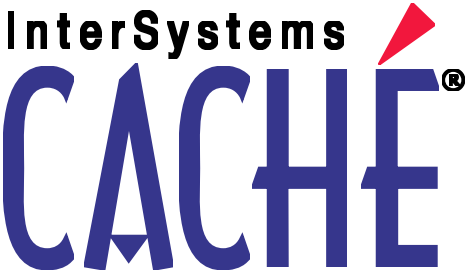


Introduction to InterSystems DeepSee

Version 2009.1
30 June 2009

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Introduction to InterSystems DeepSee
Caché Version 2009.1 30 June 2009
Copyright © 2009 InterSystems Corporation
All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.

InterSystems CACHÉ® and InterSystems ENSEMBLE®

Caché WEBLINK, Distributed Cache Protocol, M/SQL, M/NET, and M/PACT are registered trademarks of InterSystems Corporation.

INTERSYSTEMS, InterSystems HEALTHSHARE™, InterSystems DEEPSEE™ and InterSystems TRAKCARE™

InterSystems Jalapeño Technology, Enterprise Cache Protocol, ECP, and InterSystems Zen are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Customer Support
Tel: +1 617 621-0700
Fax: +1 617 374-9391
Email: support@InterSystems.com

Table of Contents

About This Book **1**

1 Introduction to DeepSee **3**
1.1 Purpose 3
1.2 Architecture 3
1.3 Pivot Tables 5
1.3.1 Content 6
1.3.2 Display Options 6
1.3.3 Alerts 7
1.3.4 Chart Properties 8
1.3.5 User Options 8
1.4 Dashboards 9
1.4.1 Available Dashboard Elements 9
1.4.2 Other Dashboard Properties 10
1.5 Role-based Security 10
1.6 Real-time Incremental Index Updates 11

2 Logging Into DeepSee **13**

3 The DeepSee Model **15**
3.1 Introduction 15
3.2 DeepSee Cubes and BI-enabled Classes 16
3.3 Dimensions 17
3.3.1 Numeric Dimensions and Measures 18
3.3.2 Date-type Dimensions and Variants 18
3.3.3 Indexed Dimensions and Dynamic Dimensions 19
3.3.4 Compound Dimensions 19
3.4 Filters 19
3.5 Subject Areas 20
3.6 Detail Listings and Listing Fields 20
3.6.1 Listing Fields 21
3.6.2 Detail Listings 21
3.7 Measures and Computations 21
3.7.1 Measures 22
3.7.2 Computations 22
3.7.3 Automatically Created Measures and Computations 22
3.7.4 Custom Measures and Computations 23

4 Understanding the User Options **25**

4.1 Viewing Detail Listings 25
4.1.1 Demo, Part 1 25
4.1.2 Demo, Part 2 28
4.1.3 Design Options 29
4.2 Drilling Through 29
4.2.1 Demo 29
4.2.2 Design Options 33
4.3 Drilling Down 33
4.3.1 Demo 34
4.3.2 Design Options 36
4.4 Drilling Down to an Associated Pivot Table 36
4.4.1 Demo 36
4.4.2 Design Options 38

5 Components of InterSystems DeepSee 39
5.1 DeepSee Architect 39
5.1.1 Users 40
5.1.2 Use Options 40
5.1.3 A Quick Look 41
5.2 DeepSee Analyzer 42
5.2.1 Users 42
5.2.2 Use Options 42
5.2.3 A Quick Look 42
5.3 DeepSee Designer 44
5.3.1 Users 44
5.3.2 A Quick Look 45
5.4 DeepSee Connector 45
5.4.1 Users 45
5.4.2 Use Options 46
5.4.3 A Quick Look 46
5.5 Additional Modules 46
5.6 Overall Sequence 47

Index 49

List of Tables

Supporting DeepSee Modules 47

About This Book

This book introduces InterSystems DeepSee, which enables you to embed business intelligence (BI) into your applications. The purpose of this book is to describe the capabilities that you can add to your applications and to give a detailed look at the results that you can provide to your users. It defines the key concepts, introduces the DeepSee components that you use during different phases of development, and outlines how you use these components together.

This book is written primarily for implementers who are responsible for using InterSystems DeepSee. The first couple of chapters, however, may also be useful for anyone else who wants to become acquainted with InterSystems DeepSee.

This book contains the following sections:

- "Introduction to DeepSee" introduces DeepSee.
- "Logging Into DeepSee" describes how to log into DeepSee.
- "The DeepSee Model" introduces the key elements of the DeepSee Model — the elements that underlie DeepSee pivot tables.
- "Understanding the User Options" demonstrates how users can interact with pivot tables and dashboards in your applications, and it summarizes how you can customize these options.
- "Components of InterSystems DeepSee" introduces the components of InterSystems DeepSee.

For a detailed outline, see the table of contents.

For more information, see the following books:

- *Using the DeepSee Analyzer*, a guide for implementers and advanced users who want to create pivot tables to embed in applications — or who simply want to explore their data.
- *Using the DeepSee Architect*, a guide for implementers who are setting up a DeepSee model.
- *Using the DeepSee Dashboard Designer*, a guide for implementers who are creating dashboards.
- *Using the DeepSee Connector*, a guide for implementers who are using the DeepSee Connector to access externally stored data.

  Note: The DeepSee Connector is available only with Ensemble.

- *DeepSee Site Configuration and Maintenance Guide*, a guide for implementers and system administrators. This book describes how to configure and maintain a DeepSee site. It also includes a chapter on troubleshooting.

- *DeepSee User Guide*, a user manual for your end users. This book describes how to work with deployed dashboards and pivot tables.

For general information, see the *InterSystems Documentation Guide*.

1

Introduction to DeepSee

This chapter provides an introduction to InterSystems DeepSee.

Be sure to consult *InterSystems Supported Platforms* for information on system requirements.

1.1 Purpose

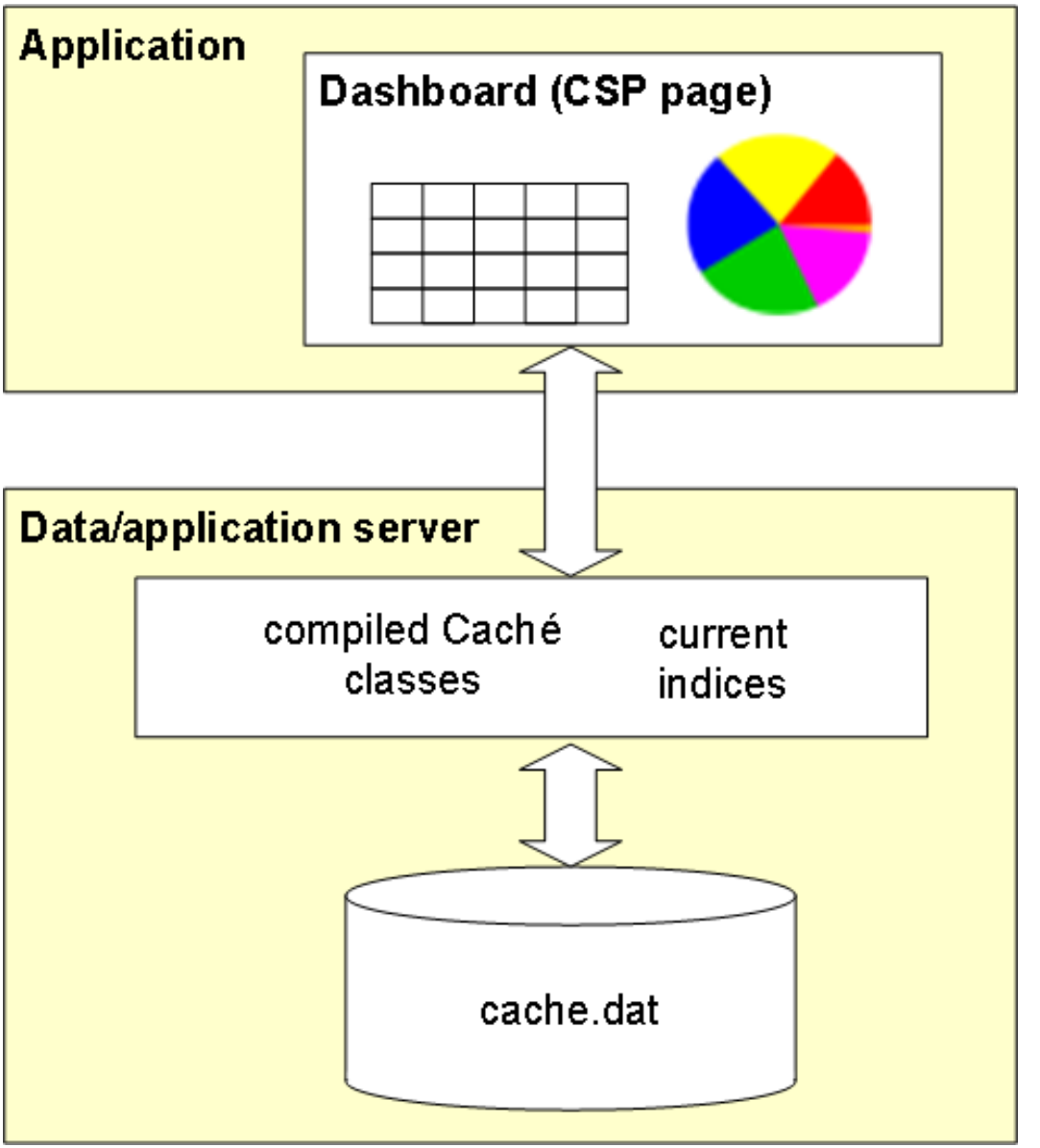
The purpose of InterSystems DeepSee is to enable you to embed business intelligence (BI) into your applications so that your users can ask and answer sophisticated questions of their data. Your application can include any or all of the following:

- DeepSee *pivot tables*, which are interactive, drillable displays of data, generally designed for specific user roles or for specific areas of your user interface.
- More complex *dashboards*, which include pivot tables, various controls such as date choosers, and speedometers.
- An analysis screen (the DeepSee Analyzer) for advanced end users, to enable them to explore their data in an open-ended manner.

1.2 Architecture

In the typical scenario, dashboards are added to an existing application and make use of the same data that the rest of the application uses. (It is also possible for the dashboards to execute against a shadow of the data, if that is necessary for some reason.)

Most dashboards contain at least one pivot table. Pivot tables work by accessing BI-enabled class definitions (which constitute the DeepSee model) and data indices. The following shows a generic example:



When you define dimensions, DeepSee adds the needed indices to the data. In order for DeepSee to function, these indices must be current. As part of implementation, you will set up procedures to rebuild the indices when necessary (as you would with any other indices in your applications).

Note the following technical points about the DeepSee indices:

- These indices are bitmap indices, but they use an implementation of bitmap indices that predates the standard Caché bitmap indices. This means that they are suitable only for use by DeepSee.
- These indices are case sensitive.
- Nulls are not indexed. DeepSee does, however, provide options for handling nulls.

1.3 Pivot Tables

Pivot tables are central to InterSystems DeepSee; they select and aggregate data and display it in an interactive format.

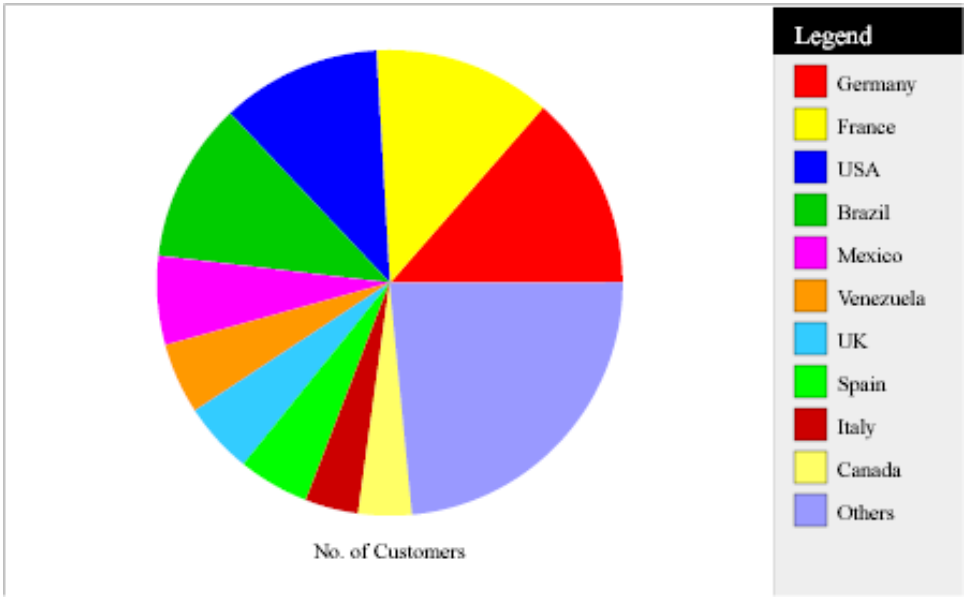
In InterSystems DeepSee, the phrase *pivot table* is a generic phrase that can mean a tabular display of data, a graph of the same data, or a combination format. We start by looking at pivot tables in table format.

The following figure shows an example pivot table, in table format. It shows how many orders were placed in different countries, grouped by product category and order year.

Country/category breakdown Level 0 of 0

		2005	2006	2007
Ship Country	Category	Orders	Orders	Orders
Argentina	Beverages		1	5
	Condiments		1	2
	Confections		4	3
	Dairy Products		2	3
	Grains/Cereals			1
	Produce		2	2
	Seafood		2	2
Austria	Beverages	5	8	7
	Condiments	4	9	3
	Confections	2	8	3
	Dairy Products	5	13	8
	Grains/Cereals	2	6	6
	Meat/Poultry	1	7	2

Similarly, a pivot table can be displayed as a graph, in a wide variety of styles:



1.3.1 Content

You can control the content of pivot tables in all the following ways:

- You define the dimensions, measures, and computations (apart from a few predefined measures and computations).
- You choose the dimensions for the rows and columns.
- You choose the measures and computations shown in the main body of the table.
- You control whether null rows and null columns are included, and you can specify how null values are replaced, if wanted.

1.3.2 Display Options

In addition to controlling the content, you can control the appearance of any given pivot table as follows:

- You can specify the default sorting, maximum number of rows and columns, and maximum number of rows per page.
- You can rank the items, control how ranking is performed (top or bottom, number of items), specify which number is used for ranking (number of customers), and control whether to display the remainder or not.

- You control the use of titles. You have control over every caption.
- You can specify the default font and colors used in the table.
- You can resize the columns, and you can specify the row height.
- You can combine multiple tables, by placing them side-by-side, by stacking them vertically, or by nesting them. The ability to link tables visually gives you greater control over layout.
- You can hide the pivot header (which contains buttons for the users).
- You can add labels (empty, labelled rows or columns) to group data visually or for other reasons.

1.3.3 Alerts

A pivot table can have alerts. For example:

	2005	2006	2007
Ship Country	Orders	Orders	Orders
Argentina	NONE	6	10
Austria	8	21	11
Belgium	2	7	10
Brazil	13	42	28
Canada	4	17	9
Denmark	3	11	4
Finland	4	13	5
France	15	39	23
Germany	24	64	34
Ireland	5	10	4
Italy	3	15	10
Mexico	9	12	7
Norway	1	2	3
Poland	1	2	4
Portugal	4	7	2

This pivot table includes the following two alerts:

- If the value in a cell is 0 (or less), display the cell in red and display the text `NONE` instead of the value.
- If the value in a cell is in the range 1–3, inclusive, display the cell in red.

The definition of an alert includes the following parts:

- The cells to consider
- A value range, which can be open-ended
- An alert display style

An alert examines each data cell in the pivot table, checks whether the value in the cell falls into the alert range, and if so, applies the alert display style to that cell.

When a pivot table is displayed, the system uses each alert, one at a time, starting with the first alert. For a given cell, if two alerts apply to the cell, the latter alert takes precedence.

1.3.4 Chart Properties

For a pivot table shown in chart format, you can control the relevant properties from the previous sections as well as features such graph type, and the color and fill patterns used in the graph.

1.3.5 User Options

As noted earlier, a pivot table is an interactive display of data. Within a pivot table, a user can do all of the following:

- View detail listings (described in the following chapter)
- Drill through and drill down to other views of the data (described in the following chapter)
- Toggle the display of the drill options list, which is used when drilling
- Export to Microsoft Excel, Excel charts, Microsoft Word, text files
- Print the data
- Resize columns, hide columns, and display hidden columns
- Sort the data by clicking on column headers
- Display the data in table format, chart format or both
- Control the type of chart, position of the legend, and so on
- Control fonts, colors, and other display options
- Control whether null rows and null columns are displayed
- Add alerts

1.4 Dashboards

To include a pivot table in your application, you package it up in a dashboard, which is a CSP page. You then include this CSP page in your application.

A dashboard can be as simple or as complex as needed. First let us consider a rather simple one:

To use this dashboard, a user can click one or more shippers in the list; this updates the pivot table instantly.

1.4.1 Available Dashboard Elements

A dashboard can include some or all of the following elements:

- Pivot tables
- Detail listings
- Speedometers, which display KPIs (key performance indicators) that you have defined
- Labels and images
- Buttons, drop-down lists, text boxes, check boxes, and other user interface controls to use as filters for the pivot tables or detail listings

- DeepSee components such as the Analyzer

You control the size and position of all these elements.

1.4.2 Other Dashboard Properties

Also, you can specify properties of the dashboard such as the following:

- Background color or background image
- Dimensions of the dashboard
- Fonts used in the dashboard
- Password needed to display the dashboard, for added security
- Refresh timer

1.5 Role-based Security

In DeepSee (as in the rest of Caché and Ensemble), access to data and to functions is secured by roles.

The roles to which a user belongs determine the user's access or ability as follows:

- Roles control whether the user can use pivot tables at all.
- Roles determine the subject areas that the user can see. A subject area is a sub-cube that corresponds to a specific set of data, and some or all of the dimensions defined for that area. For example, a subject area could consist of all lab tests, all lab tests for a specific hospital, or all lab tests of a given type. Subject areas are discussed later in this book.
- Roles determine the detail listings that the user can see. A detail listing is a dialog box that displays selected data fields for the source data that underlies a given number shown in a pivot table. Detail listings are discussed more fully later in this book.
- Roles control whether the user can use the different components of DeepSee, and if so, which options are available. For information on these components, see the chapter "Components of InterSystems DeepSee."

1.6 Real-time Incremental Index Updates

Rebuilding indices, in general, can be time-consuming, so DeepSee provides tools to update the indices incrementally.

First, if you enable incremental updates, DeepSee does the following: If a data change affects a record in the base BI-enabled class itself, DeepSee automatically detects that change and updates the indices for that record. This occurs because the base BI-enabled class inherits from %BI.Adaptor, which provides the method %OnAfterSave() and the triggers **HyperDelete**, **HyperInsert**, and **HyperUpdate**. When an object is saved, Caché runs the method %OnAfterSave(), which updates the DeepSee indices for this object. Similarly, when an record is deleted, inserted, or updated via SQL, Caché fires the corresponding trigger, which updates the DeepSee indices for this object.

Your DeepSee model can use data in other classes; if that data changes, it is the responsibility of your application to determine the IDs of the affected objects in the base class and then call the **zzBuildOne()** method of the base class to update the DeepSee indices for those classes.

2

Logging Into DeepSee

To log into DeepSee:

1. Click the InterSystems Launcher.

   When you do so, the system displays a menu.

2. Click **DeepSee**.

   If you have not yet specified a namespace, the system displays a page that prompts you for a namespace.

   Otherwise, the system displays the DeepSee login page.

3. If you are prompted for a namespace, type the name of the namespace you want to work in and then click **Logon to DeepSee**.

   The system then displays the DeepSee login page.

4. On the DeepSee login page, enter a DeepSee username and password. For example, you can use the username `demo` with the password `demo`.

5. For **Role**, select `demo`.

6. Click **Login**.

3

The DeepSee Model

This chapter discusses the key elements of the DeepSee Model — the elements that underlie DeepSee pivot tables. It discusses the following topics:

- An introduction to the terms
- Cubes and BI-enabled classes
- Dimensions
- Filtering
- Subject areas
- Detail listings and listing fields
- Measures and computations

3.1 Introduction

Because the DeepSee concepts are interrelated, making it impossible to discuss each concept without reference to the others, it is useful for us to start with several working definitions:

- A *dimension* is a construct that we use when we group the source data. A dimension (such as Country) has *members*. Each member, in turn, corresponds to a specific set of records in the source data. We group the records separately for each member.
- A *measure* is a number or date that is based on numeric or date values selected from the source data. A measure aggregates those values and represents them with a single value, which is shown in a pivot table.
- A *listing field* is a source data field, with a user-friendly name assigned to it if necessary.

- A *detail listing* is a view of the source data that underlies a given member displayed in a pivot table. It shows the values of the listing fields for all these source records.

3.2 DeepSee Cubes and BI-enabled Classes

In computing, the word *cube* is often defined in terms of its implementation via "pre-aggregated arrays" or "pre-aggregated tables." However, in Caché and Ensemble, everything is ultimately stored in globals; this includes all data, all class definitions, all routines — and all DeepSee cubes. So for our purposes, a more generic definition is needed. In a general sense, a cube is a construct that is optimized to support OLAP-type queries (such as the ones shown in the preceding chapter).

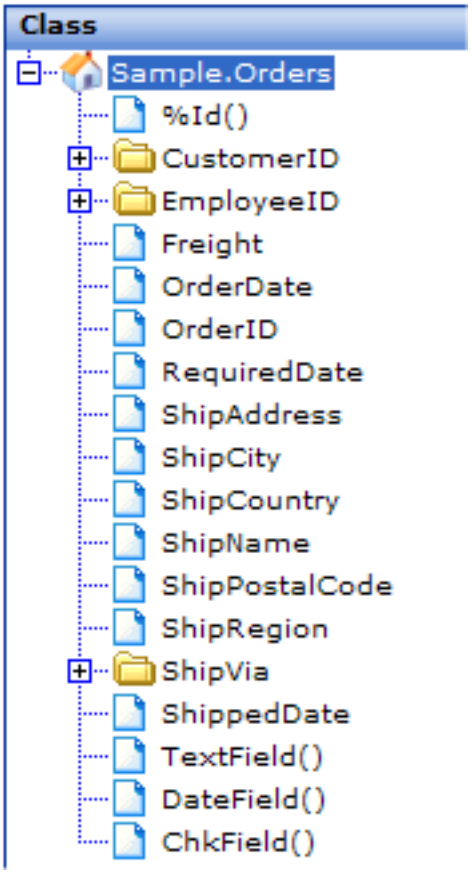
In InterSystems DeepSee, to build a cube, you enable a set of Caché classes to work with DeepSee; we say that you make the classes *BI-enabled* (by analogy with "XML-enabled" and other such terms). To do this, you include a specific class (%BI.Adaptor) as a superclass of your existing classes.

Actually, you only need to BI-enable a single class. When you do so, that class and all classes related to it are automatically included in the generated cube. Consider a simple example:

- The `MyApp.Customer` class defines customers.
- The `MyApp.Shipper` class defines shippers.
- The `MyApp.Orders` class defines orders placed by customers and shipped by various shippers. It includes the properties `Customer`, which is of type `MyApp.Customer`, and `Shipper`, which is of type `MyApp.Shipper`.

If you BI-enable `MyApp.Orders`, and recompile and rebuild the classes, you automatically create a DeepSee cube that contains all the data from these three classes (and any other related classes).

In general, any property of these classes can be used as either a dimension, a listing field, or both. When you define dimensions and listing fields, you work in the DeepSee Architect, which displays these classes in a hierarchical format as follows:



The root item and the items shown as folders represent the class definitions. The other items are properties. You can expand the folders, which contain more properties and possibly more class definitions.

The root item is special; this class is the *base class* in this cube. The base class determines the meaning of the predefined `Count` measure, as we will see later in this chapter.

In practice, you might BI-enable multiple classes, in order to explore multiple areas. For example, you might perform analysis on patients, as well as on doctors and departments.

3.3 Dimensions

A *dimension* is a construct used when we group the source data. A dimension (such as Country) has *members*. Each member, in turn, corresponds to a specific set of records in the source data. When a pivot table uses a given dimension, it groups the records separately for each member of that dimension.

As you saw earlier, when a pivot table uses multiple dimensions, it provides the appropriate breakdowns for all members of these dimensions.

When you define dimensions, you can select any property within the BI-enabled classes to use as a dimension. You can specify a user-friendly name, rather than the current property name, as shown in the following example:

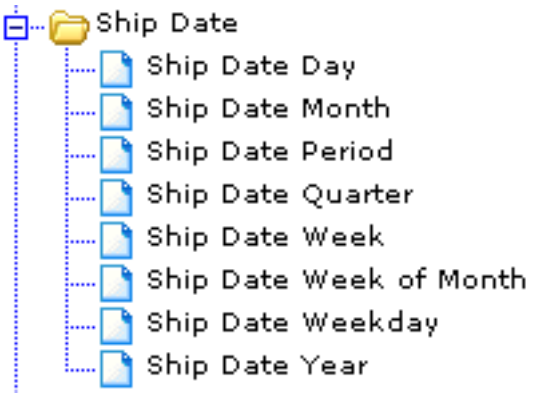
ID	Dimension Name	Property	Data Type
10001	Customer	CustomerID.CompanyName	Values
10002	Staff		Values
10003	Freight	Freight	Number
10004	Ship City	ShipCity	Values
10005	Ship Country	ShipCountry	Values
10006	Ship Region	ShipRegion	Values
10007	Shipper	ShipVia.CompanyName	Values
10008	Ship Date	ShippedDate	Date
10009	Order Date	OrderDate	Date

3.3.1 Numeric Dimensions and Measures

If you use a numeric-valued property (such as Freight) as a dimension, DeepSee automatically creates a measure based on that property. That is, you can use the property either as a dimension or as a measure, depending on your needs.

3.3.2 Date-type Dimensions and Variants

If you use a date-valued property (such as Order Date) as a dimension, DeepSee automatically provides a set of variants that use different parts of the date. For example:



3.3.3 Indexed Dimensions and Dynamic Dimensions

By default, dimensions are indexed. That is, when you define a dimension, an index is added to the class that contains the property that you are using. The data in the index must be up-to-date in order for the dimension to behave correctly.

You can also create dynamic dimensions, which are not indexed.

3.3.4 Compound Dimensions

You can create more highly aggregated dimensions based on members of existing dimensions. These are known as *compound dimensions*. The definition of a compound dimension consists of the following set of information:

- Name of the compound dimension.
- Set of members of this dimension. Each member definition consists of a member name and an expression that selects records from the database.

For example, if you already have a Country dimension, you can create a Continent dimension. The definition of, for example, the Europe member would be an expression that selects records for Austria, Belgium, Denmark, and so on.

For information on expressions that select records, see the following section, "Filters."

3.4 Filters

You use *filters* in the definition of pivot tables and in many other places.

A filter is a boolean expression that specifies which source records to select. Creating a filter in DeepSee does not require any SQL. You simply choose a dimension and then choose members of that dimension — from the members that are *currently* present in your data. No separate lookup table is needed. The resulting filter might look like the following:

```
[Country = Argentina]
```

This filter selects only data associated with Argentina.

You can create more advanced filters that use logical operators, other comparison operators, and even embedded Caché ObjectScript expressions.

3.5 Subject Areas

In InterSystems DeepSee, you do not work directly with the BI-enabled classes. Instead, you work with *subject areas*, which are sub-cubes.

The definition of a subject area consists of the following details:

- Name of the subject area, in user-friendly language. These names are visible to anyone who creates pivot tables or who uses the Analyzer.
- A set of user roles that have access to this subject area.
- A subset (or all) of the dimensions defined for these BI-enabled classes.

  For example, if a given user role has access to data from only one country, that user does not need the Country and Continent dimensions. For another example, there might be some breakdowns that only certain users should be able to view; only those users would have access to the associated dimensions.
- An optional filter.

  When a pivot table runs, if the pivot table is unfiltered, it retrieves all the data in the subject area. It is therefore important to filter the subject areas appropriately.

You can have as many subject areas as you need.

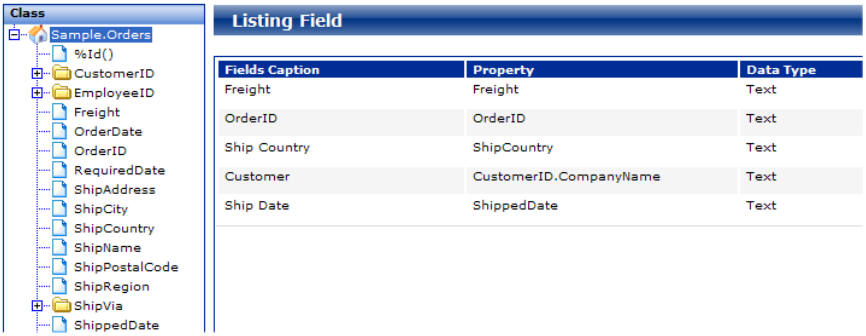
3.6 Detail Listings and Listing Fields

As shown in the previous chapter, a *detail listing* displays source data (that is, non-aggregated data) for a given member shown in a pivot table. The fields in a detail listing are known as *listing fields*.

3.6.1 Listing Fields

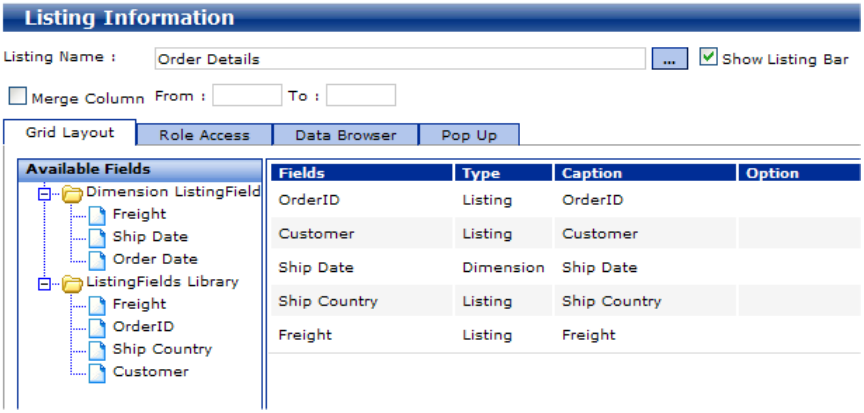
All dimensions of the BI-enabled classes are automatically available as listing fields.

Any other property of these classes can *also* be used as listing field, depending on your DeepSee model. For example, the following shows the definitions of five listing fields:



3.6.2 Detail Listings

You can define as many detail listings as you need. The definition of a detail listing indicates which listing fields to use, what captions to use, and the order to display them. For example:



The definition also specifies which user roles can see the detail listing.

3.7 Measures and Computations

The main body of a pivot table (the white area) displays measures and computations.

3.7.1 Measures

Measures refer to source data and aggregate it. That is, a given cell in a pivot table corresponds to specific records in the database. If that cell displays a measure, the measure uses a specific field in those records and gets all the values of that field (one value in each of the associated records). It then aggregates those values and displays the result.

DeepSee provides a set of aggregation functions, including sum, minimum and maximum, average, first date and last date.

Measures can be used only as columns.

3.7.2 Computations

In contrast to measures, *computations* refer to data in other cells of the pivot table and use that data to compute a related number.

The expression for a computation uses Caché ObjectScript. A large selection of functions, macros, and special variables is available, and you can use any Caché ObjectScript expression. For example, you can create computations that do the following:

- Compute the difference between two columns or two rows
- Compute a cumulative total
- Divide one column by another and display as a percentage
- Use If-Then-Else logic to determine which number to display
- Compute the total value shown in another column or row (or use this total within a more complex calculation)

Computations can be used as rows or as columns.

3.7.3 Automatically Created Measures and Computations

DeepSee automatically creates the following measures and computations:

- The `Count` measure is simply a count of the records in the base class (in this case, `User.Orders`), automatically aggregated as a sum.
- In addition, a cube automatically provides a measure corresponding to each numeric dimension. Any such measure is automatically aggregated as a sum.
- The `Total` computation provides a total. When you use this computation as a row, a given cell displays the sum of all values in the column, as shown in the example after this list. When you use this computation as a column, a given cell displays the sum of all values in the row.

- The `Average` computation displays the average value of a column or row, in the same way. That is, when you use it as a row, a given cell displays the average of all values in the column. When you use it as a column, a given cell displays the average of all values in the row.
- The `Forecast` computation displays a forecast value, in the same way. The forecast is calculated as the moving average of the numbers.

3.7.4 Custom Measures and Computations

You can create any custom measures and computations you need. You can define them within a subject area, which makes their initial definitions available to anyone working in that subject area. Or you can create them locally within a given pivot table.

You could create custom measures and computations such as the following:

- Running totals.
- Backlog calculations.
- Average number of days since a particular event.
- Number of associated items. For example, a custom measure could indicate the number of lab tests or the number of types of tests.
- Expressions using Caché ObjectScript.

4

Understanding the User Options

It is important to understand how users can work with pivot tables and dashboards in your applications. This chapter presents demos that show key user options, and it summarizes how you can customize these options. It discusses the following topics:

- How the user can view a detail listing that shows the source data associated with a number shown in a pivot table
- How the user can drill through and see a breakdown of a particular cell in a pivot table
- How the user can drill down and see a more complex breakdown of a particular cell in a pivot table
- How the user can drill down to an associated pivot table for an even more detailed look at the data

4.1 Viewing Detail Listings

When your users view the embedded pivot tables in your applications, they can display *detail listings*, dialog boxes that show source data for a given cell in a pivot table.

4.1.1 Demo, Part 1

1. In this demo, we start with the following pivot table:

Orders by country

Ship Country	Orders	Freight
Argentina	16	598.58
Austria	40	7,391.50
Belgium	19	1,280.14
Brazil	83	4,877.79
Canada	30	2,198.09
Denmark	18	1,396.19
Finland	22	910.89
France	77	4,238.45
Germany	122	11,282.28

2. We right-click in the row for Argentina and then click **Listing to Screen**.

   The system then displays the detail listing in another window, in front of the pivot table. For example:

Order Details

OrderID	Customer	Ship Date	Ship Country	Freight
10409	Océano Atlántico Ltda.	14 Jan 2006	Argentina	29.83
10448	Rancho grande	24 Feb 2006	Argentina	38.82
10521	Cactus Comidas para llevar	02 May 2006	Argentina	17.22
10531	Océano Atlántico Ltda.	19 May 2006	Argentina	8.12
10716	Rancho grande	27 Oct 2006	Argentina	22.57
10782	Cactus Comidas para llevar	22 Dec 2006	Argentina	1.1
10819	Cactus Comidas para llevar	16 Jan 2007	Argentina	19.76
10828	Rancho grande	04 Feb 2007	Argentina	90.85
10881	Cactus Comidas para llevar	18 Feb 2007	Argentina	2.84
10898	Océano Atlántico Ltda.	06 Mar 2007	Argentina	1.27
10916	Rancho grande	09 Mar 2007	Argentina	63.77
10937	Cactus Comidas para llevar	13 Mar 2007	Argentina	31.51
10958	Océano Atlántico Ltda.	27 Mar 2007	Argentina	49.56
10986	Océano Atlántico Ltda.	21 Apr 2007	Argentina	217.86
11019	Rancho grande		Argentina	3.17
11054	Cactus Comidas para llevar		Argentina	.33

Page 1 of 1 Record 1 of 16

   Notice the following:

   - In this case, the detail listing shows information for 16 orders, which are the orders placed by customers based in Argentina.

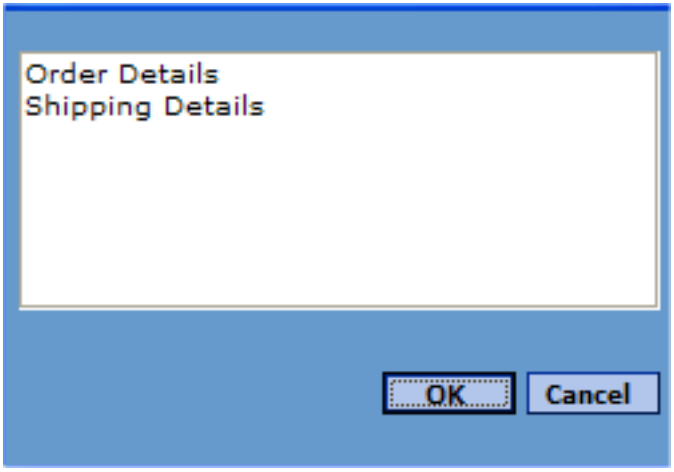
- The detail listing shows the order ID and other basic information that is stored at the lowest level.

3. We click the X in the upper right of the detail listing to dismiss it.

   Next we look at another detail listing.

4. We right-click in the Argentina row and select **Temp. Listing Selection**. This menu item is displayed because there are multiple possible detail listings.

   The system displays the following dialog box:



5. We have already seen `Order Details`, so we click the other detail listing and then click **OK**.

   The system then displays the following:

Shipping Details

OrderID	Shipper	Ship Country	Ship Date	Freight
10409	Speedy Express	Argentina	14 Jan 2006	29.83
10448	United Package	Argentina	24 Feb 2006	38.82
10521	United Package	Argentina	02 May 2006	17.22
10531	Speedy Express	Argentina	19 May 2006	8.12
10716	United Package	Argentina	27 Oct 2006	22.57
10782	Federal Shipping	Argentina	22 Dec 2006	1.1
10819	Federal Shipping	Argentina	16 Jan 2007	19.76
10828	Speedy Express	Argentina	04 Feb 2007	90.85
10881	Speedy Express	Argentina	18 Feb 2007	2.84
10898	United Package	Argentina	06 Mar 2007	1.27
10916	United Package	Argentina	09 Mar 2007	63.77
10937	Federal Shipping	Argentina	13 Mar 2007	31.51
10958	United Package	Argentina	27 Mar 2007	49.56
10986	United Package	Argentina	21 Apr 2007	217.86
11019	Federal Shipping	Argentina		3.17
11054	Speedy Express	Argentina		.33

Page 1 of 1 Record 1 of 16

Notice that the fields are different in this detail listing, which is meant to give information related to shipping the orders.

4.1.2 Demo, Part 2

As noted earlier, a detail listing displays source data for the dimension member or members to which that cell applies. In the examples so far, the pivot table used only one dimension (ship country). More complex pivot tables use multiple dimensions. This means that the detail listing you see depends on *where* in the row you started.

1. In this demo, we start with the following pivot table:

Orders by country & year

	2005		2006		2007	
Ship Country	Orders	Freight	Orders	Freight	Orders	Freight
Argentina			6	117.66	8	477.42
Austria	7	1,132.63	20	3,730.26	11	2,190.51
Belgium	2	57.57	7	460.48	10	762.09
Brazil	13	1,221.49	39	2,146.28	29	1,342.47
Canada	4	136.31	17	1,687.76	8	303.44
Denmark	2	40.33	11	810.95	4	526.47
Finland	4	197.43	13	636.11	5	77.35

2. We right-click in the Argentina row in either of the 2006 columns and then click **Listing to Screen**. The system then displays the following:

Order Details

OrderID	Customer	Ship Date	Ship Country	Freight
10409	Océano Atlántico Ltda.	14 Jan 2006	Argentina	29.83
10448	Rancho grande	24 Feb 2006	Argentina	38.82
10521	Cactus Comidas para llevar	02 May 2006	Argentina	17.22
10531	Océano Atlántico Ltda.	19 May 2006	Argentina	8.12
10716	Rancho grande	27 Oct 2006	Argentina	22.57
10782	Cactus Comidas para llevar	22 Dec 2006	Argentina	1.1

|< < Page 1 of 1 > >| |< < Record 1 of 6 > >|

Notice that we see data only for 2006.

3. We click the X in the upper right of the detail listing to dismiss it.

4. Now we right-click in the Argentina row in either of the 2007 columns and then click **Listing to Screen**. The system then displays the following:

Order Details

OrderID	Customer	Ship Date	Ship Country	Freight
10819	Cactus Comidas para llevar	16 Jan 2007	Argentina	19.76
10828	Rancho grande	04 Feb 2007	Argentina	90.85
10881	Cactus Comidas para llevar	18 Feb 2007	Argentina	2.84
10898	Océano Atlántico Ltda.	06 Mar 2007	Argentina	1.27
10916	Rancho grande	09 Mar 2007	Argentina	63.77
10937	Cactus Comidas para llevar	13 Mar 2007	Argentina	31.51
10958	Océano Atlántico Ltda.	27 Mar 2007	Argentina	49.56
10986	Océano Atlántico Ltda.	21 Apr 2007	Argentina	217.86

Page 1 of 1 Record 1 of 8

As always, the detail listing shows the records for the numbers you are examining; in this case, that means the orders sold during 2007 to customers based in Argentina.

4.1.3 Design Options

As the developer, you have the following design options for detail listings.

- You can create as many detail listings as needed.
- You specify which user roles can access each detail listing.
- You specify which detail listing is the default for any given pivot table.
- When you create a detail listing, you specify the fields, the order in which they are shown, and the captions used on the columns.

4.2 Drilling Through

When your users view the embedded pivot tables in your applications, they can choose an area and *drill through* to see a breakdown that uses a different dimension. This action is best explained by an example.

4.2.1 Demo

1. In this demo, we start with the following pivot table:

Top ten countries

Ship Country	Orders	No. of Customers
USA	122	14
Germany	122	13
Brazil	83	9
France	77	11
UK	56	7
Venezuela	46	4
Austria	40	2
Sweden	37	2
Canada	30	3
Mexico	28	5
Others	189	23

2. We right-click in the table and select **Show Measure List**.

   (Instead we could click the drill list button ( ) in the header of the pivot table.)

The pivot table is then redisplayed with the drill options list on the left:

Ship Country	Orders	No. of Customers
USA	122	14
Germany	122	13
Brazil	83	9
France	77	11
UK	56	7
Venezuela	46	4
Austria	40	2
Sweden	37	2
Canada	30	3
Mexico	28	5
Others	189	23

The drill options list includes all dimensions to which we can drill through (in this pivot table), as well as any drill-downs we can use. Drill-downs are discussed in the next section.

3. Next we drag `Ship City` from the drill options list and drop it in any cell in the `UK` row. The pivot table is then redisplayed as follows:

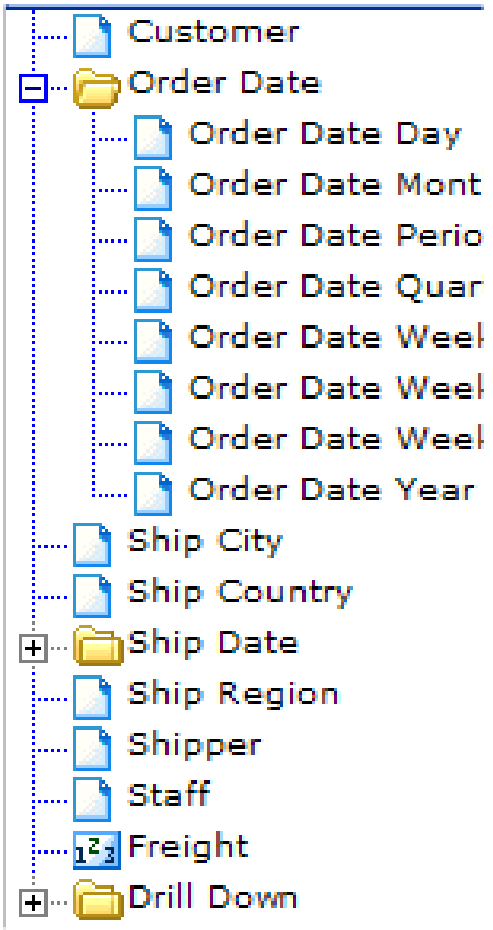
Ship City	Orders	No. of Customers
London	33	5
Colchester	13	1
Cowes	10	1

This pivot table now shows a close-up for the UK, showing the ship cities for this country. Notice the following in the header:

- `Level 1 of 1` indicates that we are now in the first level of drilling.
- `Ship Country = UK` indicates that we are viewing data for the UK.

- > Ship City indicates that we are viewing a breakdown by the Ship City dimension.

4. Now we expand the Order Date folder in the drill options list:



5. Then we drag Order Date Year from the drill options list to any cell in the London row. The system displays the following:

This pivot table now shows a close-up for London, showing the order years for customers in this city. By displaying `Level 2 of 2`, the header indicates that we are now in the second level of drilling.

6. We click the left arrow in the pivot table banner, which redisplays the UK close-up:

7. Now we drag `Shipper` from the drill options list and drop it onto the `London` row. This time we see a breakdown by shipper, of the orders that were sent to London:

Of the seven total customers in London, four used Federal Shipping, three used Speedy Express, and four used United Package. That is, some customers used more than one shipper.

4.2.2 Design Options

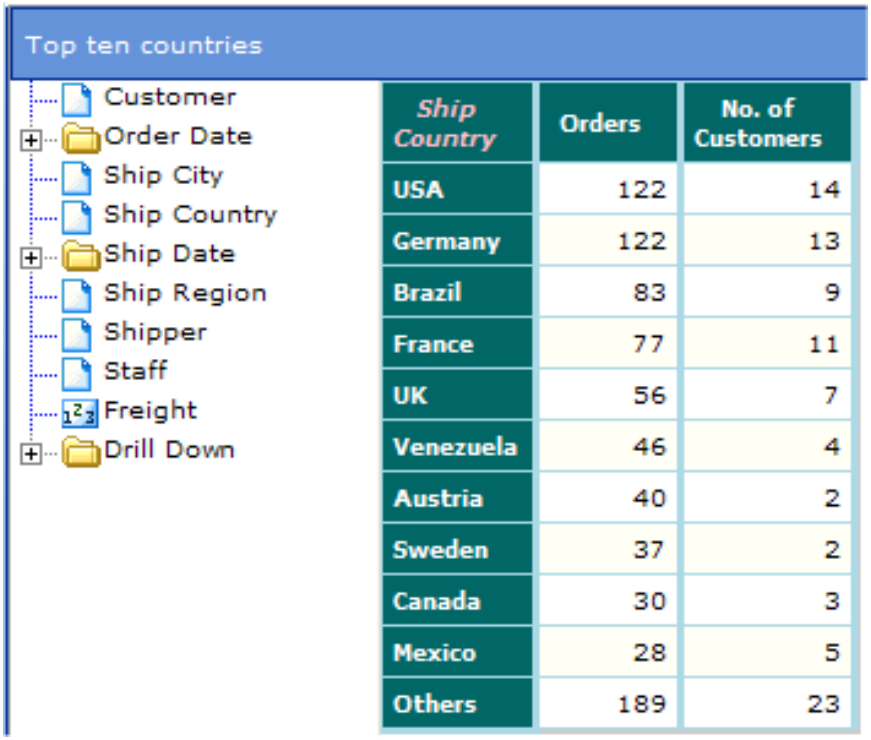
As the developer, you specify the dimensions to list in the drill options list. You can also control the order in which they are shown, so that you can place more typical options at the top and less commonly used ones at the bottom (instead of displaying them alphabetically as shown here). Each pivot table can have its own drill options, as is appropriate for the context — or you can provide global options for use in all pivot tables.

4.3 Drilling Down

When your users view the embedded pivot tables in your applications, they can also *drill down*, which is slightly different. This action is also best explained by an example:

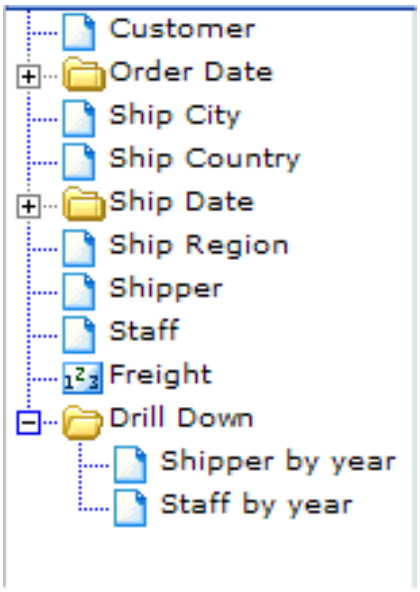
4.3.1 Demo

1. In this demo, we start with the following pivot table:

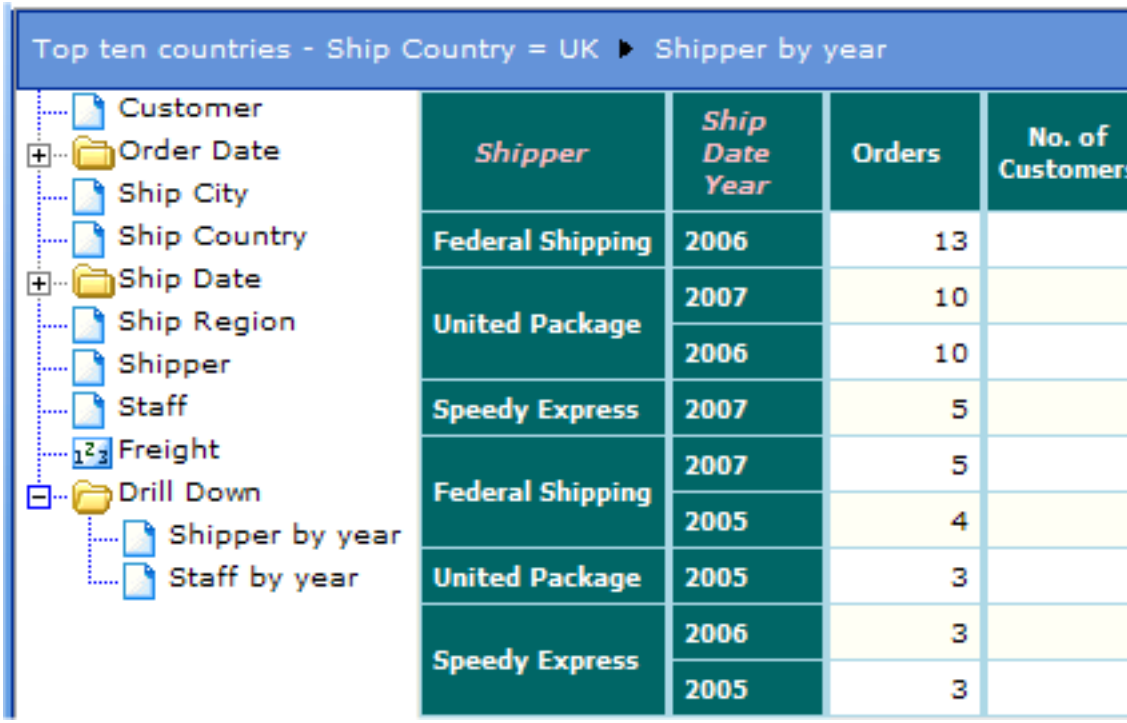


Ship Country	Orders	No. of Customers
USA	122	14
Germany	122	13
Brazil	83	9
France	77	11
UK	56	7
Venezuela	46	4
Austria	40	2
Sweden	37	2
Canada	30	3
Mexico	28	5
Others	189	23

2. We expand the `Drill Down` folder in the drill options list:



3. We drag `Shipper by year` and drop it within the `UK` row of the pivot table. Then we see the following:

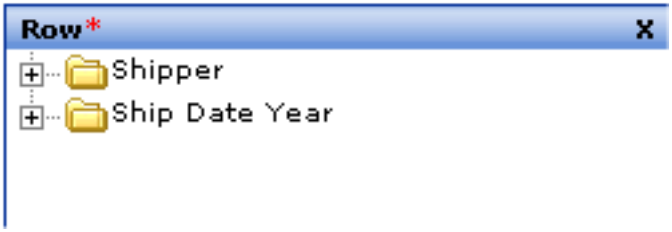


Shipper	Ship Date Year	Orders	No. of Customers
Federal Shipping	2006	13	
United Package	2007	10	
	2006	10	
Speedy Express	2007	5	
Federal Shipping	2007	5	
	2005	4	
United Package	2005	3	
Speedy Express	2006	3	
	2005	3	

In comparison to drilling through, we see a breakdown by multiple dimensions — in this case, shipper and year. Notice that the drill-down we used (`Shipper by year`) was given a name that indicated the dimensions it uses.

4.3.2 Design Options

As the developer, you create the drill-downs. The definition of a drill-down consists essentially of an ordered list of dimensions, like this:



The definition here means that the data is first broken down by shipper, and within shipper is broken down by year. The user cannot see the definition of a drill-down, which is why it is useful to provide a name that indicates the dimensions used.

You also specify where a drill-down is available — either for all pivot tables available to the user or for a specific pivot table.

4.4 Drilling Down to an Associated Pivot Table

Users can also drill down to an associated pivot table that you have added to the right-click menu.

In contrast to the previous types of drilling, here the user sees measures that were not included in the original pivot table.

4.4.1 Demo

1. In this demo, we start with the following pivot table:

Top ten countries

Ship Country	Orders	No. of Customers
USA	122	14
Germany	122	13
Brazil	83	9
France	77	11
UK	56	7
Venezuela	46	4
Austria	40	2
Sweden	37	2
Canada	30	3
Mexico	28	5
Others	189	23

2. We right-click Canada and then click `Drill Down to Shippers by ship year (By Row)`.

   The system then displays the following in another window in front of the original pivot table:

Shippers by ship year

	2005	2006	2007
Shipper	Count	Count	Count
United Package	4	5	
Speedy Express		2	2
Federal Shipping		10	6

From the original pivot table, we know that 30 orders were ordered by customers in Canada. This pivot table shows us when and how those orders were shipped. One order has not yet been shipped, which is why we see only 29 orders here.

4.4.2 Design Options

Any pivot table can have any number of associated pivot tables that can be launched this way. These associations are completely under your control, as the developer. You have complete control over the definitions of the pivot tables themselves, as discussed earlier.

5

Components of InterSystems DeepSee

This chapter introduces the components of InterSystems DeepSee. It discusses the following topics:

- DeepSee Architect, which you use to define the DeepSee model for use in the DeepSee Analyzer.
- DeepSee Analyzer, which you use to define pivot tables or to simply explore the data.
- DeepSee Designer, which you use to define dashboards, in particular dashboards that embed pivot tables.
- DeepSee Connector, which you use to access data in external database or flat files, in order to generate classes (and data-loading methods) that serve as the basis of the DeepSee model. This component is available only with Ensemble, and it is not needed in all cases.
- Additional, supporting modules.
- An overview of how you use these components in combination.

5.1 DeepSee Architect

The purpose of DeepSee Architect is to define the building blocks needed to create pivot tables and dashboards. When you are done, you will have the following major elements:

- BI-enabled classes.
- Dimensions based on properties of those classes.

- Automatically created measures based on properties. You can add more measures with the DeepSee Analyzer.
- Detail listing fields based on properties.
- Detail listings associated with different user roles, as needed.
- Subject areas associated with different user roles, as needed.

5.1.1 Users

The Architect is meant for use by implementers, primarily application partners of InterSystems. In order to use the Architect most effectively, it is useful to have a good understanding of the application that you are extending.

5.1.2 Use Options

The Architect lets you work with and modify any BI-enabled classes, which are any classes that extend the DeepSee superclass (%BI.Adaptor).

The first step depends on where your data is now:

- If you have Caché classes, you can open them from within Architect. This automatically adds the DeepSee superclass (%BI.Adaptor) to the superclass list.
- If you are creating a new application, you can create it in Studio in the same way you create any other Caché application. For any class that needs it, you include the DeepSee superclass (%BI.Adaptor) in the superclass list.
- If you are starting with an existing Caché application that does not use Caché class definitions, map the existing globals to class definitions and then add the DeepSee superclass (%BI.Adaptor) to the superclass list.
- If your data is stored outside of Caché, you can first use the DeepSee Connector to browse the data and generate class definitions and their corresponding data-loading methods.

In all cases, the Architect automatically has access to any classes that extend %BI.Adaptor.

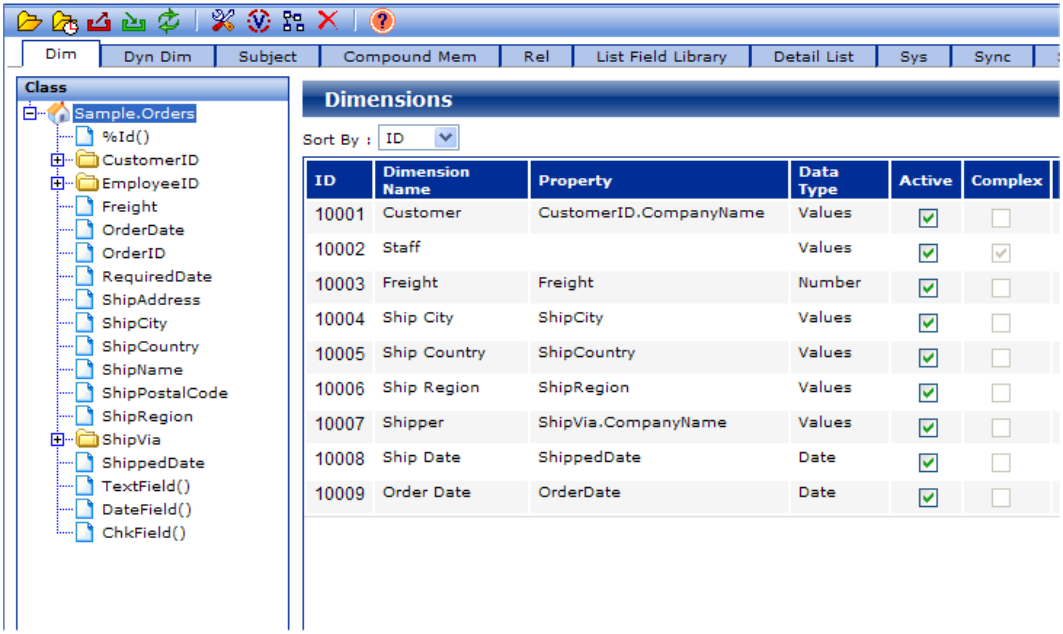
Once you have a BI-enabled class, you use the Architect to create dimensions, measures, and the other model elements based on this class. Some changes directly affect your class definition, but other changes are written to globals and are not visible in the class definition.

The Architect accesses and modifies the same class definition that you see directly in the Studio. This means that if you make a change in one location and save it, that change is immediately available in the other.

For convenience, you can recompile and rebuild the classes as needed from Architect.

5.1.3 A Quick Look

The Architect looks like this:



In this example, we are looking at the BI-enabled class `Sample.Orders` and its related classes. We can use all properties of these classes when we define the DeepSee model. As you can see, nine dimensions have already been created.

In general, to use the Architect, you first open a BI-enabled class. The Architect displays this class and related classes as a hierarchy, as you can see in the previous example. Tabs in the Architect display different model elements related to that class and its related classes. In most cases, you use the following tabs:

- The **Dim** tab displays the dimensions defined within these classes.
- The **Dyn. Dim** tab displays the dynamic (non-indexed) dimensions defined within these classes.
- The **Subject** tab displays the subject areas defined for these classes and specifies the roles that can access them.
- The **Compound Member** tab displays the compound dimensions defined within these classes. These dimensions may or may not be indexed.
- The **List Field Library** tab displays the listing fields defined for these classes, apart from the dimensions (which are automatically used as listing fields and which are not shown on this tab).

- The **Detail List** tab displays the detail listings defined for these classes and specifies the roles that can access them.

The other tabs are for special cases and are not described here.

5.2 DeepSee Analyzer

The purpose of DeepSee Analyzer is to define pivot tables, by using the building blocks that were created within the Architect.

5.2.1 Users

The Analyzer is intended for implementers and advanced users. It is not necessary to be acquainted with OLAP terminology and concepts; nor is any knowledge of SQL needed.

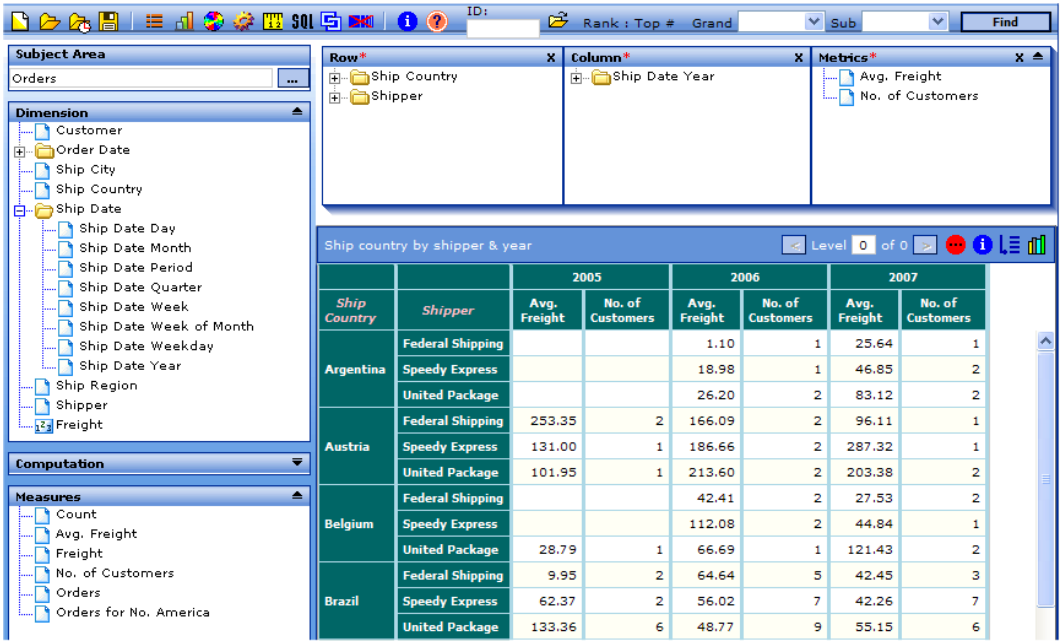
5.2.2 Use Options

In most cases, the Analyzer is used to create pivot tables, but you can also use it simply to explore the data, without ever saving a pivot table.

Also, the Analyzer can be packaged within a dashboard and can thus be provided as an embedded user interface within your application, for the appropriate set of users.

5.2.3 A Quick Look

The Analyzer looks like this:



The area on the left displays the building blocks available within a subject area:

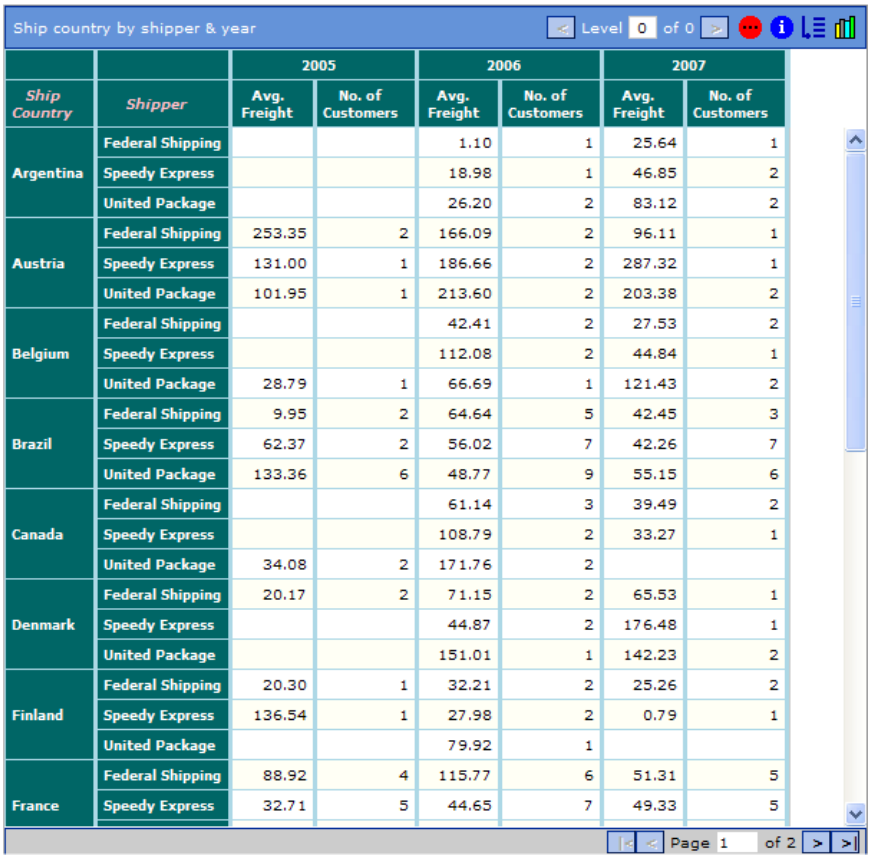
- The **Subject Area** panel shows the currently selected subject area.
- The **Dimension** panel shows the dimensions in that subject area that are available to your roles.
- The **Computation** panel shows the computations defined in this subject area. This includes predefined computations and any computations you have added.
- The **Measures** panel shows the measures defined in this subject area. This includes predefined measures and any measures you have added.
- The **Drill Down** panel (not shown) shows the drill-downs, if any, that you have defined in this subject area or for this pivot table.
- The **Others** panel (not shown) lets you add time series, scorecard rows, and labels to the currently viewed pivot table.
- The **Filter** panel (not shown) lets you define and see the filter applied to the currently viewed the pivot table.

The panels at the top define the currently displayed pivot table:

- The **Row** panel displays the dimensions used to group the data rows, as well as any labels, computations, or other elements used as rows.
- The **Column** panel displays the dimensions used to group the data columns, as well as any other elements used as columns.

- The **Metrics** panel displays the measures or computations shown in the data cells.

The bottom right area displays a fully functioning preview of the pivot table that you are currently creating or modifying, as in the following example:



Ship country by shipper & year — Level 0 of 0

Ship Country	Shipper	2005 Avg. Freight	2005 No. of Customers	2006 Avg. Freight	2006 No. of Customers	2007 Avg. Freight	2007 No. of Customers
Argentina	Federal Shipping			1.10	1	25.64	1
	Speedy Express			18.98	1	46.85	2
	United Package			26.20	2	83.12	2
Austria	Federal Shipping	253.35	2	166.09	2	96.11	1
	Speedy Express	131.00	1	186.66	2	287.32	1
	United Package	101.95	1	213.60	2	203.38	2
Belgium	Federal Shipping			42.41	2	27.53	2
	Speedy Express			112.08	2	44.84	1
	United Package	28.79	1	66.69	1	121.43	2
Brazil	Federal Shipping	9.95	2	64.64	5	42.45	3
	Speedy Express	62.37	2	56.02	7	42.26	7
	United Package	133.36	6	48.77	9	55.15	6
Canada	Federal Shipping			61.14	3	39.49	2
	Speedy Express			108.79	2	33.27	1
	United Package	34.08	2	171.76	2		
Denmark	Federal Shipping	20.17	2	71.15	2	65.53	1
	Speedy Express			44.87	2	176.48	1
	United Package			151.01	1	142.23	2
Finland	Federal Shipping	20.30	1	32.21	2	25.26	2
	Speedy Express	136.54	1	27.98	2	0.79	1
	United Package			79.92	1		
France	Federal Shipping	88.92	4	115.77	6	51.31	5
	Speedy Express	32.71	5	44.65	7	49.33	5

Page 1 of 2

5.3 DeepSee Designer

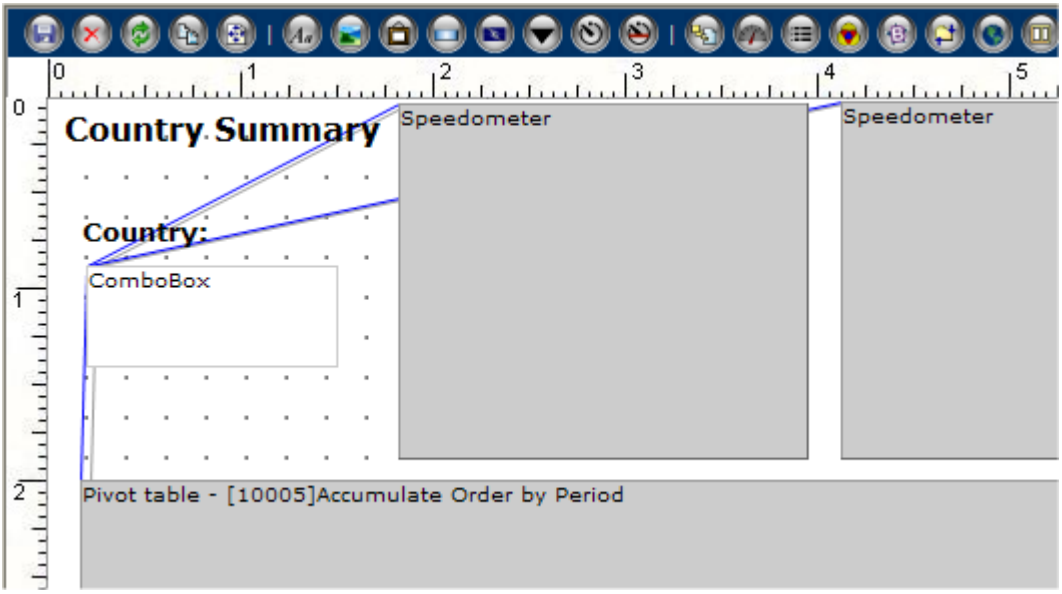
The purpose of the DeepSee Dashboard Designer is to define dashboards — in particular, dashboards that include pivot tables.

5.3.1 Users

The Designer is intended primarily for implementers.

5.3.2 A Quick Look

The Designer displays an editable version of a dashboard, along with tools you can use to make changes. For example:



The white and gray boxes represent elements displayed on the dashboard, and the blue and gray lines represent different kinds of connections between the elements. In this case, the combo box element is connected to each speedometer and to the dashboard in two ways:

- The filter specified by the combo box is applied to the associated elements.
- When the combo box is used, the associated elements are refreshed.

5.4 DeepSee Connector

The purpose of the DeepSee Connector is to access and browse externally stored data (in external database or in flat files) and to generate classes that model the data, along with data-loading methods for those classes. This component is available only with Ensemble.

5.4.1 Users

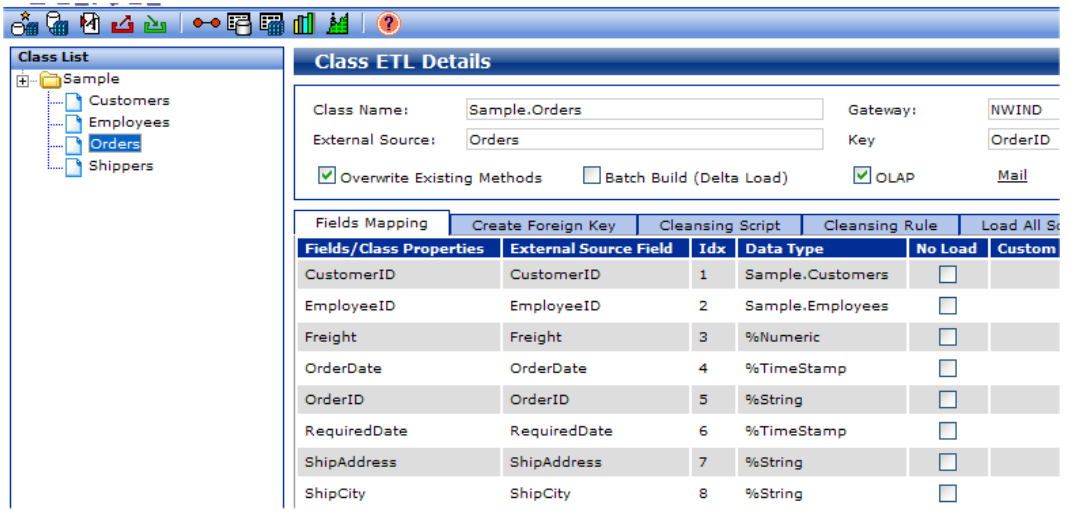
The Connector is intended primarily for implementers.

5.4.2 Use Options

You generally use the Connector in two phases. First you use it to access the external data and specify its basic structure (primary keys, data types, and so on); in this step, you create the initial classes and methods that load the data for them from the external sources. Then you use it to examine the classes, fine-tune the structure, and specify additional options such as data cleansing rules.

5.4.3 A Quick Look

The Connector looks like the following:



In general, the list at the left (Class List) shows the classes that you have created with the Connector. The area at the right shows details for the selected class. The buttons at the top launch additional wizards that can modify the currently selected classes.

5.5 Additional Modules

DeepSee provides additional, supporting modules which include the following:

Supporting DeepSee Modules

Module	Purpose	For Information
Transformations	Defines reusable transformations that you can use within the Architect.	See *Using the DeepSee Architect*.
KPI Setup	Defines KPIs (key performance indicators) that you can use within the Analyzer and the Dashboard Designer.	See *Using the DeepSee Analyzer*.
Venn Diagram Setup	Defines Venn diagrams that you can use within the Dashboard Designer.	See *Using the DeepSee Dashboard Designer*.
Scheduler	Schedule the tasks of rebuilding indices, loading data, and executing custom Caché ObjectScript.	See the *DeepSee Site Configuration and Maintenance Guide*.
Site Configuration	Specify site-wide configuration options.	
Role Maintenance	Define DeepSee roles.	
User Maintenance	Define DeepSee users.	
Shortcut Management, User Preferences	Configure the DeepSee user interface as seen by each user.	

5.6 Overall Sequence

In general, you use these components together as follows:

1. Use the Site Configuration module to define basic site options such as the locations of log files.
2. Optionally use the DeepSee Connector to access external data and generate classes.

   Also use the Transformations module to define transformations for use in the Connector.

   Otherwise, use Studio to create new classes or to BI-enable existing classes.
3. Use the DeepSee Architect to define the DeepSee model.

   Also use the Transformations module to define transformations for use in the Connector and the Architect.

   If necessary, return to the Connector and make changes there.

Both the Architect and the Connector can potentially make changes to the class definitions; both components provide a convenient option to recompile.

4. Use the DeepSee Analyzer to define pivot tables.

   Also, use the KPI Setup module to define KPIs for use in pivot tables and in dashboards.

   If necessary, return to the Architect (or less often, the Connector), make changes to the model, and recompile or rebuild if necessary.

5. Use the DeepSee Designer to create dashboards.

   Also, use the KPI Setup module and the Venn Diagram Setup module to define KPIs and Venn diagrams for use in dashboards.

   If necessary, return to the Analyzer and change or add pivot tables.

6. Use the Scheduler module to schedule the tasks of data loading (if applicable) and full index rebuilding (if necessary; see the section "Real-time Incremental Index Updates").

Index

Symbols

%BI.Adaptor, 16

A

alerts, 7
Analyzer
  displaying within dashboard, 42
  during implementation, 48
  introduction, 42
Architect
  during implementation, 47
  introduction, 39
architecture, 3

B

BI-enabled classes, 16
business intelligence (BI), 3

C

Caché ObjectScript
  in computations, 22
  in filters, 19
charts
  displaying pivot table as, 6
  properties of, 8
compound dimensions, 19
computations
  custom, 23
  overview, 22
  predefined, 22
Connector
  during implementation, 47
  introduction, 45
cubes
  in DeepSee, 16
  sub-cubes, 20

D

dashboards
  basics, 9
  creating, 44
  elements of, 9
  in your applications, 3
  other properties, 10
  packaging the Analyzer within, 42
DeepSee
  architecture, 3
  components, 39
  model, 15
  purpose, 3
Designer
  during implementation, 48
  introduction, 44
detail listing fields
  defining, 41
  overview, 21
detail listings
  context sensitivity, 28
  defining, 42
  overview, 21
  viewing, 25
dimensions
  defining, 41
  overview, 17
  used in filters, 19
display options, 6
drilling down, 33
drilling down to associated pivot table, 36
drilling through, 29

E

export options, 8

F

filters, 19

G

graphs
  displaying pivot table as, 6
  properties of, 8

I

indices, 5

M

measures
  custom, 23
  overview, 22
  predefined, 22

O

OLAP-enabled classes, 16

P

pivot tables
  alerts, 7
  content, 6
  creating, 42
  displayed as graph, 6
  display options, 6
  introduction, 5
  user options, 8
print options, 8

R

roles, 10

S

security, 10
sub-cubes, 20
subject areas, 20
  defining, 41

U

user access, 10